DETERMINAREA SOLUȚIILOR ECUAȚIILOR DIOFANTICE 2069-2080

CIRA OCTAVIAN

REZUMAT. În acest articol ne propunem să determinăm soluţii pentru ecuaţiile diofantice 2069-2080.

1. INTRODUCERE

Funcţia care asociază fiecărui număr natural n pe cel mai mic număr natural m care are proprietatea că $m!$ este multiplu lui n a fost considerată de prima dată de Lucas, [3], în anul 1883. Alţi autori care au considerat această funcţie în lucrările lor sunt: Nueberg (1887) [4], Kempner (1918) [2]. Această funcţie a fost redescoperită de Smarandache, [5], în anul 1980. Funcţia este notată de Smarandache cu S sau cu η, iar în site-ul Wolfram MathWorld, [10], se notează cu μ. În acest articol am adoptat notaţia η a lui Smarandache din lucrarea [7].

În monografia [7], Smarandache prezintă o listă de ecuaţii diofantice referitoare la funcţia η. Fie $m, n \in \mathbb{N}^*$ fixaţi şi $x, y \in \mathbb{N}$ necunoscutele. Ecuaţiile diofantice 2069-2080 cu funcţia η sunt:

(2069) $\eta(m \cdot x + n) = x$,
(2070) $\eta(m \cdot x + n) = m + n \cdot x$,
(2071) $\eta(m \cdot x + n) = x!$,
(2072) $\eta(x^m) = x^n$,
(2073) $\eta(x)^m = \eta(x^n)$,
(2074) $\eta(m \cdot x + n) = \eta(x)^y$,
(2075) $\eta(x) + y = x + \eta(y)$, unde x şi y nu sunt prime.
(2076) $\eta(x) + \eta(y) = \eta(x + y)$, unde x şi y nu sunt prime gemene. În general funcţia η nu este o funcţie aditivă.
(2077) $\eta(x + y) = \eta(x) \cdot \eta(y)$. În general funcţia η nu este o funcţie exponenţială.
(2078) $\eta(x \cdot y) = \eta(x) \cdot \eta(y)$. În general funcţia η nu este o funcţie multiplicativă.
(2079) $\eta(m \cdot x + n) = x^y$,
(2080) $\eta(x) \cdot y = x \cdot \eta(y)$, unde x şi y nu sunt prime.

Date: 03 Aprilie 2014.

2. ALGORITMUL KUMPNER

Funcţia $\eta : \mathbb{N}^* \to \mathbb{N}^*$, $\eta(n) = m$, unde m este cel mai mic număr natural care are proprietatea că n îl divide pe $m!$, (sau $m!$ este multiplu de n) este cunoscută în literatura de specialitate ca *funcţia lui Smarandache*. Valorile funcţiei, pentru $n = 1, 2, \ldots, 18$, sunt: 1, 2, 3, 4, 5, 3, 7, 4, 6, 5, 11, 4, 13, 7, 5, 6, 17, 6 obţinute cu un algoritm ce rezultă din definiţia funcţiei η, după cum urmează:

Programul 2.1.

$$\eta(n) = \left| \begin{array}{l} for\ m = 1..n \\ \quad return\ m\ if\ mod(m!, n) = 0 \end{array} \right.$$

Acest algoritm nu poate fi folosit pentru $n \geq 19$ pentru că numerele $19!$, $20!$, …sunt numere cu mai mult de 17 cifre zecimale şi în modul clasic de calcul (fără o aritmetică a preciziilor arbitrare [9]) se vor genera erori de trunchiere.

Kempner, [2], a dat un algoritm pentru a calcula pe $\eta(n)$ folosind factorizarea clasică $p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_s^{\alpha_s}$ cu numere prime a lui n, şi baza de numeraţie generalizată $(\alpha_i)_{[p_i]}$, pentru $i = \overline{1, s}$. Soluţii parţiale pentru algoritmul de calcul a lui $\eta(n)$ s-au dat mai devreme de Lucas şi Neuberg, [8] .

Prezentăm algoritmul lui Kempner, [2], de calcul al funcţiei lui Smarandache η. Pentru început definim şirul recursiv $a_{j+1} = p \cdot a_j + 1$ cu $j = 1, 2, \ldots$ şi $a_1 = 1$, unde p este un număr prim. Acest şir de numere constituie baza de numeraţie generalizată a lui p. Deoarece $a_2 = p + 1$, $a_3 = p^2 + p + 1$, …se poate demonstra prin inducţie că $a_j = 1 + p + \ldots + p^{j-1} = (p^j - 1)/(p - 1)$ pentru $\forall j \geq 2$.

Valoarea lui ν, astfel încât $a_\nu \leq \alpha < a_{\nu+1}$, este dat de formula

$$\nu = \left\lfloor \log_p \big(1 + \alpha(p-1)\big) \right\rfloor \ , \tag{2.1}$$

unde $\lfloor \cdot \rfloor$ este funcţia *parte întreagă inferioară*. Cu ajutorul algoritmului lui Euclid putem determina şirurile unice κ_i şi r_i după cum urmează

$$\alpha = \kappa_\nu \cdot a_\nu + r_\nu \ , \tag{2.2}$$

$$r_\nu = \kappa_{\nu-1} \cdot a_{\nu-1} + r_{\nu-1} \ , \tag{2.3}$$

$$\vdots \tag{2.4}$$

$$r_{\nu-(\lambda-2)} = \kappa_{\nu-(\lambda-1)} \cdot a_{\nu-(\lambda-1)} + r_{\nu-(\lambda-1)} \ , \tag{2.5}$$

$$r_{\nu-(\lambda-1)} = \kappa_{\nu-\lambda} \cdot a_{\nu-\lambda} \ . \tag{2.6}$$

adică, până când restul $r_{\nu-\lambda} = 0$. La fiecare pas κ_i este partea întreagă a raportului r_i/a_i şi r_i este restul împărţirii. De exemplu pentru primul pas avem $\kappa_\nu = \lfloor \alpha/a_\nu \rfloor$ şi $r_\nu = \alpha - \kappa_\nu \cdot a_\nu$. Atunci avem

$$\eta(p^\alpha) = (p-1)\alpha + \sum_{i=\nu}^{\lambda} \kappa_i \; . \tag{2.7}$$

În general pentru

$$n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_s^{\alpha_s} \; , \tag{2.8}$$

valoarea funcţie η este dată de formula:

$$\eta(n) = \max\left\{\eta(p_1^{\alpha_1}), \eta(p_2^{\alpha_2}), \ldots, \eta(p_s^{\alpha_s})\right\} \; , \tag{2.9}$$

formulă ce o datorăm lui Kempner [2].

Remarca 2.2. Dacă $n \in \mathbb{N}^*$ are descompunerea în produs de numere prime (2.8), unde p_i sunt numere prime astfel încât $p_1 < p_2 < \ldots < p_s$, iar $s \geq 1$, atunci algoritmul lui Kempner de calcul al funcţiei η este

$$\eta(n) = \max\left\{p_1 \cdot \left(\alpha_{1_{[p_1]}}\right)_{(p_1)}, \; p_2 \cdot \left(\alpha_{2_{[p_2]}}\right)_{(p_2)}, \; \ldots, \; p_s \cdot \left(\alpha_{s_{[p_s]}}\right)_{(p_s)}\right\} \; , \tag{2.10}$$

unde prin $\left(\alpha_{[p]}\right)_{(p)}$ înţelegem că numărul α este "scris" în baza de numeraţie generalizată p (notată $\alpha_{[p]}$) şi este "citit" în baza de numeraţie p (notat $\beta_{(p)}$, unde $\beta = \alpha_{[p]}$), [6, p. 39].

3. Programe

Programul 3.1. Funcţia care numără cifrele în baza de numeraţie p a lui n

$$ncb(n,p) := \left| \begin{array}{l} return\ ceil(\log(n,p))\ if\ n > 1 \\ return\ 1\ otherwise \end{array} \right.$$

unde funcţia utilitară Mathcad $ceil(\cdot)$ este funcţia parte întreagă superioară.

Programul 3.2. Program pentru generarea a bazei de numeraţie generalizată a lui p (notată de Smarandache $[p]$) pentru un număr cu m cifre

$$a(p,m) := \left| \begin{array}{l} for\ i \in 1..m \\ \quad a_i \leftarrow \dfrac{p^i - 1}{p-1} \\ return\ a \end{array} \right.$$

Programul 3.3. Program de generarea a bazei de numeraţie p (notată de Smarandache (p)) pentru a putea scrie numărul α

$$b(\alpha, p) := \left| \begin{array}{l} return\ (1)\ if\ p = 1 \\ for\ i \in 1..ncb(\alpha, p) \\ \quad b_i \leftarrow p^{i-1} \\ return\ b \end{array} \right.$$

Programul 3.4. Programul determină cifrele bazei de numeraţie generalizată $[p]$ pentru numărul n

$$Nbg(n, p) := \left| \begin{array}{l} m \leftarrow ncb(n, p) \\ a \leftarrow a(p, m) \\ return\ (1)\ if\ m = 0 \\ for\ i \in m..1 \\ \quad \left| \begin{array}{l} c_i \leftarrow trunc\left(\dfrac{n}{a_i}\right) \\ n \leftarrow \mod(n, a_i) \end{array} \right. \\ return\ c \end{array} \right.$$

Programul 3.5. Program pentru funcţia Smarandache

$$\eta(n) := \left| \begin{array}{l} return\ "Err.\ n\ nu\ este\ intreg"\ if\ n \neq trunc(n) \\ return\ "Err.\ n < 1"\ if\ n < 1 \\ return\ (1)\ if\ n = 1 \\ f \leftarrow Fa(n) \\ p \leftarrow f^{\langle 1 \rangle} \\ \alpha \leftarrow f^{\langle 2 \rangle} \\ for\ k = 1..rows(p) \\ \quad \eta_k \leftarrow p_k \cdot Nbg(\alpha_k, p_k) \cdot b(\alpha_k, p_k) \\ return\ \max(\eta) \end{array} \right.$$

Acest program apelează programul $Fa(n)$ de factorizare cu numere prime. Programul foloseşte remarca 2.2 a lui Smarandache referitor la algoritmul Kempner.

Programul 3.6. Programul de generarea numerelor prime până la L cu ajutorul ciurului lui Atkin.

$$Atkin(L) := \left| \begin{array}{l} eprim_L \leftarrow 0 \\ \lambda \leftarrow floor(\sqrt{L}) \\ for\ j \in 1..\lambda \\ \quad \left| \begin{array}{l} for\ k \in 1..ceil\left(\dfrac{\sqrt{L - j^2}}{2}\right) \\ \quad \left| \begin{array}{l} n \leftarrow 4k^2 + j^2 \\ m \leftarrow mod(n, 12) \\ eprim_n \leftarrow \neg eprim_n\ if\ n \leq L \wedge (m = 1 \vee m = 5) \end{array} \right. \end{array} \right. \end{array} \right.$$

$$
\begin{array}{l}
\quad\quad for\ k \in 1..ceil\left(\dfrac{\sqrt{L-j^2}}{3}\right)\\
\quad\quad\quad n \leftarrow 3k^2 + j^2\\
\quad\quad\quad eprim_n \leftarrow \neg eprim_n\ if\ n \leq L \wedge mod(n,12) = 7\\
\quad\quad for\ k \in j+1..ceil\left(\dfrac{\sqrt{L+j^2}}{3}\right)\\
\quad\quad\quad n \leftarrow 3k^2 - j^2\\
\quad\quad\quad eprim_n \leftarrow \neg eprim_n\ if\ n \leq L \wedge mod(n,12) = 11\\
\quad for\ k \in 5,7,..\lambda\\
\quad\quad for\ j = 1,3,..\dfrac{L}{k^2}\ if\ eprim_k\\
\quad\quad\quad eprim_{j\cdot k^2} \leftarrow 0\\
\quad eprim_1 \leftarrow 2\\
\quad eprim_2 \leftarrow 3\\
\quad j \leftarrow 3\\
\quad for\ n \in 3,5,..L\\
\quad\quad if\ eprim_n\\
\quad\quad\quad prime_j \leftarrow n\\
\quad\quad\quad j \leftarrow j+1\\
\quad return\ prime
\end{array}
$$

Programul 3.7. Programul de factorizare a unui număr natural.

Acest program foloseşte vectorul numerelor prime p care este generat cu ciurul lui Atkin, [1], cel mai rapid program de generare a numerelor prime până la limita precizată. Apelul ciurului lui Atkin se face cu secvenţa:

$$t_0 = time(0)$$

$$p := Atkin(10^8) \quad last(p) = 5761455 \quad p_{last(p)} = 99999989$$

$$t_1 = time(1) \quad (t_1 - t_0)sec = 82.88sec\ .$$

$$
Fa(m) := \left|
\begin{array}{l}
return\ ("m = "\ m\ "> ca\ ultimul\ p^2")\ if\ m > (p_{last(p)})^2\\
j \leftarrow 1\\
k \leftarrow 0\\
f \leftarrow (1\ 1)\\
while\ m \geq p_j\\
\quad if\ \ \mod(m, p_j) = 0\\
\quad\quad k \leftarrow k+1\\
\quad\quad m \leftarrow \dfrac{m}{p_j}\\
\quad otherwise
\end{array}\right.
$$

$$\begin{array}{l} \quad\quad f \leftarrow stack[f,(p_j,k)]\ if\ k>0 \\ \quad\quad j \leftarrow j+1 \\ \quad\quad k \leftarrow 0 \\ f \leftarrow stack[f,(p_j,k)]\ if\ k>0 \\ return\ submatrix(f,2,rows(f),1,2) \end{array}$$

Programul 3.8. Program de calcul al valorilor funcţiei η pentru numerele naturale până la 10^6.

$$ValFS(N) := \left| \begin{array}{l} VFS_1 \leftarrow 1 \\ for\ n \in 2..N \\ \quad VFS_n \leftarrow \eta(n) \\ return\ VFS \end{array} \right.$$

Apelul acestui program va genera un vector care va fi salvat ca un fişier de tip *prn*. Astfel vom putea folosi acest fişier pentru rezolvarea ecuaţiilor diofantice. Cu ajutorul funcţie Mathcad $time$ s-a contorizat timpul de generare al vectorului VFS.

$$N := 10^6\ \ t_0 = time(0)\ \ VFS := ValFS(N)\ \ t_1 : time(1)$$

$$(t_1 - t_0)\cdot sec = \text{"}1:7:32.623\text{"}\cdot hhmmss$$

$$WRITEPRN(\text{"}VFS.prn\text{"}) := VFS\ \ last(VFS) = 1000000$$

Pentru informare se afişează ultimul indice al vectorului VFS.

4. Rezolvarea ecuaţiilor diofantice

4.1. **Ecuaţia (2069).** Pentru căutarea soluţiilor vom citi fişierul $VFS.prn$ cu ajutorul funcţiei Mathcad $READPRN$

$$VFS := READPRN(\text{"}...\backslash VFS.prn\text{"})\ \ last(VFS) = 1000000$$

unde comanda $last(VSF)$ ne indică ultimul indice al vectorului VFS.

Generarea fişierului VFS odată şi citirea fişierului generat în documente Mathcad care determină soluţii ale ecuaţiilor diofantice face ca să rezulte o mare economie de timp de execuţie a programului de căutare a soluţiilor.

Programul 4.1. Programul de determinare a soluţiilor ecuaţiei (2069).

$$Ed2069(a_m,b_m,a_n,b_n,a_x,b_x) := \left| \begin{array}{l} S \leftarrow (\text{"}m\text{" "}n\text{" "}x\text{"}) \\ u \leftarrow last(VFS) \\ for\ m \in a_m..b_m \\ \quad for\ n \in a_n..b_n \\ \quad\quad for\ x \in a_x..b_x \\ \quad\quad\quad \eta \leftarrow m\cdot x + n \\ \quad\quad\quad q \leftarrow \eta \le u \wedge VFS_\eta = x \end{array} \right.$$

$$\left| \begin{array}{l} \quad \left| S \leftarrow stack[S,(m\ n\ x)]\ if\ q \right. \\ return\ S \end{array} \right.$$

Apelul programului se face prin secvenţa:

$$a_m := 2\ b_m := 10 \quad a_n := 1\ b_n := 10 \quad a_x := 1\ b_x := 16$$

Numărul total de cazuri verificate este:

$$(b_m - a_m + 1)(b_n - a_n + 1)(b_x - a_x + 1) = 1440\ .$$

Apelul programului Ed2069:

$$t_0 : time(0)\ \ Sol := Ed2069(a_m, b_m, a_n, b_n, a_x, b_x)\ \ t_1 := time(1)$$

Timul de execuţie în secunde şi numărul de soluţii rezultă din:

$$(t_1 - t_0) \cdot s = 0.011 \cdot s \qquad rows(Sol) - 1 = 36$$

Pentru $m \in \{2, 3, \ldots, 10\}$, $n \in \{1, 2, \ldots, 10\}$ şi $x \in \{1, 2, \ldots, 16\}$ cele 36 de soluţii ale ecuaţiei diofantice $\eta(m \cdot x + n) = x$ sunt:

m	n	x	m	n	x	m	n	x
2	4	4	4	7	7	7	5	5
2	4	6	4	8	4	7	5	10
2	5	5	4	10	5	7	6	6
2	5	10	4	10	10	7	7	7
2	6	6	5	4	4	7	8	8
2	7	7	5	5	5	8	5	15
2	9	9	5	6	6	8	7	7
2	10	5	5	7	7	8	9	9
3	5	5	5	9	9	9	7	7
3	7	7	6	9	6	9	10	10
3	7	14	6	10	5	10	7	14
3	8	8	7	3	6	10	10	5

Valoarea maximă a soluţiilor x este 15.

4.2. **Ecuaţia (2070).** Pentru căutarea soluţiilor vom citi fişierul $VFS.prn$ cu ajutorul funcţiei Mathcad $READPRN$

$$VFS := READPRN(\text{"}...\backslash VFS.prn\text{"})\ \ last(VFS) = 1000000\ .$$

Programul 4.2. Programul de determinare a soluţiilor ecuaţiei (2070).

$$Ed2070(a_m, b_m, a_n, b_n, a_x, b_x) := \left| \begin{array}{l} S \leftarrow (\text{"}m\text{" "}n\text{" "}x\text{"}) \\ u \leftarrow last(VFS) \\ for\ m \in a_m..b_m \\ \quad for\ n \in a_n..b_n \\ \quad\quad for\ x \in a_x..b_x \end{array} \right.$$

$$\begin{array}{l} \quad\quad \left| \begin{array}{l} \eta \leftarrow m \cdot x + n \\ q \leftarrow \eta \leq u \wedge VFS_\eta = m + n \cdot x \\ S \leftarrow stack[S, (m\ n\ x)]\ if\ q \end{array} \right. \\ return\ S \end{array}$$

Apelul programului se face prin secvenţa:

$$a_m := 2\ b_m := 20 \quad a_n := 1\ b_n := 20 \quad a_x := 1\ b_x := 16$$

Numărul total de cazuri verificate este:

$$(b_m - a_m + 1)(b_n - a_n + 1)(b_x - a_x + 1) = 7220\ .$$

Apelul programului Ed2070:

$$t_0 : time(0)\ \ Sol := Ed2070(a_m, b_m, a_n, b_n, a_x, b_x)\ \ t_1 := time(1)$$

Timul de execuţie în secunde şi numărul de soluţii rezultă din:

$$(t_1 - t_0) \cdot s = 0.853 \cdot s \qquad rows(Sol) - 1 = 14$$

Pentru $m \in \{2, 3, \ldots, 20\}$, $n \in \{1, 2, \ldots, 20\}$ şi $x \in \{2, 3, \ldots, 20\}$ cele 14 de soluţii ale ecuaţiei diofantice $\eta(m \cdot x + n) = x$ sunt:

m	2	4	4	6	6	8	10	12	12	14	16	18	18	20
n	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x	4	2	6	4	8	6	12	10	14	12	18	16	20	18

Valoarea maximă a soluţiilor x este 20.

4.3. **Ecuaţia (2071).** Pentru căutarea soluţiilor vom citi fişierul $VFS.prn$ cu ajutorul funcţiei Mathcad $READPRN$

$$VFS := READPRN(\text{"...}\backslash VFS.prn\text{"})\ \ last(VFS) = 1000000\ .$$

Programul 4.3. Programul de determinare a soluţiilor ecuaţiei (2071).

$$Ed2071(a_m, b_m, a_n, b_n, a_x, b_x) := \left| \begin{array}{l} S \leftarrow (\text{"m" "n" "x"}) \\ u \leftarrow last(VFS) \\ for\ m \in a_m..b_m \\ \quad for\ n \in a_n..b_n \\ \quad\quad for\ x \in a_x..b_x \\ \quad\quad\quad \left| \begin{array}{l} \eta \leftarrow m \cdot x + n \\ q \leftarrow \eta \leq u \wedge VFS_\eta = x! \\ S \leftarrow stack[S, (m\ n\ x)]\ if\ q \end{array} \right. \\ return\ S \end{array} \right.$$

Apelul programului se face prin secvenţa:

$$a_m := 2\ b_m := 15 \quad a_n := 1\ b_n := 15 \quad a_x := 1\ b_x := 19$$

Numărul total de cazuri verificate este:

$$(b_m - a_m + 1)(b_n - a_n + 1)(b_x - a_x + 1) = 3990 .$$

Apelul programului Ed2071:

$$t_0 : time(0) \;\; Sol := Ed2071(a_m, b_m, a_n, b_n, a_x, b_x) \;\; t_1 := time(1)$$

Timul de execuţie în secunde şi numărul de soluţii rezultă din:

$$(t_1 - t_0) \cdot s = 0.02 \cdot s \quad rows(Sol) - 1 = 24$$

Pentru $m \in \{2, 3, \ldots, 15\}$, $n \in \{1, 2, \ldots, 15\}$ şi $x \in \{1, 2, \ldots, 19\}$ cele 24 de soluţii ale ecuaţiei diofantice $\eta(m \cdot x + n) = x$ sunt:

$$\begin{pmatrix} m \\ n \\ x \end{pmatrix} = \begin{pmatrix} 2 & 2 & 2 & 3 & 3 & 4 & 4 & 5 & 5 & 7 & 8 & 9 & 10 \\ 3 & 10 & 12 & 7 & 9 & 4 & 6 & 1 & 3 & 15 & 12 & 9 & 6 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{pmatrix.}$$

$$\begin{matrix} 10 & 11 & 11 & 11 & 12 & 12 & 13 & 13 & 14 & 14 & 15 \\ 15 & 3 & 12 & 15 & 9 & 12 & 6 & 9 & 3 & 6 & 3 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{matrix}\right)$$

Valoarea maximă a soluţiilor x este 3.

4.4. **Ecuaţia (2072).** Pentru căutarea soluţiilor vom citi fişierul $VFS.prn$ cu ajutorul funcţiei Mathcad $READPRN$

$$VFS := READPRN(\text{"...}\backslash VFS.prn\text{"}) \;\; last(VFS) = 1000000 .$$

Programul 4.4. Programul de determinare a soluţiilor ecuaţiei (2072).

```
Ed2072(a_m, b_m, a_n, b_n, a_x, b_x) := | S ← ("m" "n" "x")
                                        | u ← last(VFS)
                                        | for m ∈ a_m..b_m
                                        |   for n ∈ a_n..b_n
                                        |     for x ∈ a_x..b_x
                                        |       | η ← x^m
                                        |       | q ← η ≤ u ∧ VFS_η = x^n
                                        |       | S ← stack[S, (m n x)] if q
                                        | return S
```

Apelul programului se face prin secvenţa:

$$a_m := 2 \; b_m := 9 \quad a_n := 2 \; b_n := 9 \quad a_x := 2 \; b_x := 10$$

Numărul total de cazuri verificate este:

$$(b_m - a_m + 1)(b_n - a_n + 1)(b_x - a_x + 1) = 576 .$$

Apelul programului Ed2072:

$$t_0 : time(0) \;\; Sol := Ed2072(a_m, b_m, a_n, b_n, a_x, b_x) \;\; t_1 := time(1)$$

Timul de execuţie în secunde şi numărul de soluţii rezultă din:

$$(t_1 - t_0) \cdot s = 0.11 \cdot s \quad rows(Sol) - 1 = 12$$

Pentru $m \in \{2, 3, \ldots, 9\}$, $n \in \{2, 3, \ldots, 9\}$ şi $x \in \{1, 2, \ldots, 10\}$ cele 12 de soluţii ale ecuaţiei diofantice $\eta(x^m) = x^n$ sunt:

$$\begin{pmatrix} m \\ n \\ x \end{pmatrix} = \begin{pmatrix} 2 & 3 & 3 & 4 & 5 & 5 & 6 & 6 & 6 & 7 & 7 & 7 \\ 2 & 2 & 2 & 2 & 2 & 3 & 2 & 2 & 3 & 2 & 2 & 3 \\ 2 & 2 & 3 & 3 & 5 & 2 & 4 & 5 & 2 & 4 & 7 & 2 \end{pmatrix}$$

Valoarea maximă a soluţiilor x este 7.

4.5. **Ecuaţia (2073).** Pentru căutarea soluţiilor vom citi fişierul $VFS.prn$ cu ajutorul funcţiei Mathcad $READPRN$

$$VFS := READPRN(\text{"...\textbackslash VFS.prn"}) \;\; last(VFS) = 1000000 \;.$$

Programul 4.5. Programul de determinare a soluţiilor ecuaţiei (2073).

```
Ed2071(a_m, b_m, a_n, b_n, a_x, b_x) := | S ← ("m" "n" "x")
                                        | u ← last(VFS)
                                        | for m ∈ a_m..b_m
                                        |   for n ∈ a_n..b_n
                                        |     for x ∈ a_x..b_x
                                        |       | q ← x^n ≤ u ∧ (VFS_x)^m = VFS_{x^n}
                                        |       | S ← stack[S, (m n x)] if q
                                        | return S
```

Apelul programului se face prin secvenţa:

$$a_m := 2 \; b_m := 9 \quad a_n := 2 \; b_n := 9 \quad a_x := 2 \; b_x := 25$$

Numărul total de cazuri verificate este:

$$(b_m - a_m + 1)(b_n - a_n + 1)(b_x - a_x + 1) = 1536 \;.$$

Apelul programului Ed2073:

$$t_0 : time(0) \;\; Sol := Ed2073(a_m, b_m, a_n, b_n, a_x, b_x) \;\; t_1 := time(1)$$

Timul de execuţie în secunde şi numărul de soluţii rezultă din:

$$(t_1 - t_0) \cdot s = 0.014 \cdot s \quad rows(Sol) - 1 = 20$$

Pentru $m \in \{2, 3, \ldots, 9\}$, $n \in \{2, 3, \ldots, 9\}$ şi $x \in \{2, 3, \ldots, 25\}$ cele 20 de soluţii ale ecuaţiei diofantice $\eta(x)^m = \eta(x^n)$ sunt:

$$\begin{pmatrix} m \\ n \\ x \end{pmatrix} = \left(\begin{matrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 5 & 5 & 5 & 5 & 6 & 6 \\ 2 & 2 & 3 & 6 & 3 & 6 & 8 & 24 & 5 & 8 & 10 & 15 & 4 & 5 \end{matrix}\right.$$

$$\left.\begin{matrix} 2 & 2 & 2 & 3 & 3 & 3 \\ 6 & 7 & 7 & 5 & 6 & 7 \\ 10 & 4 & 7 & 2 & 2 & 2 \end{matrix}\right)$$

Valoarea maximă a soluţiilor x este 24.

BIBLIOGRAFIE

1. A. O. L. Atkin and D. J. Bernstein, *Prime Sieves Using Binary Quadratic Forms*, Math. Comp. **73** (2004), 1023–1030.
2. A. J. Kempner, *Miscellanea*, Amer. Math. Monthly **25** (1918), 201–210.
3. E. Lucas, *Question nr. 288*, Mathesis **3** (1883), 232.
4. J. Neuberg, *Solutions de Questions Proposees, Question nr. 288*, Mathesis **7** (1887), 68–69.
5. F. Smarandache, *A Function in Number Theory*, Analele Univ. Timişoara, Ser. St. Math. **43** (1980), 79–88.
6. ______, *Asupra unor noi funcţii în teoria numerelor*, Universitatea de Stat Moldova, Chişinău, Republica Moldova, 1999.
7. ______, *Noi funcţii în teoria numerelor*, Universitatea de Stat Moldova, Chişinău, Republica Moldova, 1999.
8. J. Sondow and E. W. Weisstein, *Smarandache Function*, From MathWorld – A Wolfram Web Resource, http://mathworld.wolfram.com/, 2014.
9. D. Uznanski, *Arbitrary Precision*, From MathWorld–A Wolfram Web Resource, http://mathworld.wolfram.com/ArbitraryPrecision.html, 2014.
10. E. W. Weisstein, *Smarandache Function*, From MathWorld–A Wolfram Web Resource, http://mathworld.wolfram.com/DickmanFunction.html, 2014.

"AUREL VLAICU" UNIVERSITY OF ARAD